SharkTeam

SharkTeam: Analysis of the HashFlow Attack Incident

2023-06-15

SharkTeam, a leading blockchain security service team, offers smart contract audit services for developers. To satisfy the demands of different clients, thesmart contract audit services provide both manual auditing and automated auditing.

We implement almost 200 auditing contents that cover four aspects: high-level language layer, virtual machine layer, blockchain layer, and business logiclayer, ensuring that smart contracts are completely guaranteed and Safe.

On June 14, 2023, Beijing time, HashFlow fell victim to a hacker attack, resulting in an estimated profit of around $600,000 for the attackers.

SharkTeam conducted a prompt technical analysis of the incident and has summarized security measures to be taken as a precautionary approach. It is our hope that this incident serves as a lesson for future projects, contributing to the strengthening of security defenses within the blockchain industry.

1. Incident analysis

Attacker address: *0xBDf38B7475Ff810325AA39e988fb80E0aA007E84*
Attack contract: *0xDDb19a1Bd22C53dac894EE4E2FBfdB0A06769216*
Attacked contract: *0x79cdFd7Bc46D577b95ed92bcdc8abAba1844Af0c*
Attack transactions:
0xdedda493272b6b35660b9cc9070d2ea32ee61279b821184ff837e0a5752f4042
0xb08f6d3fc70b95223cfffc2c905d9c0467a589e5f652cd193e5c00b4ad329b990x08b5
f35076beb363a7206b8f9b4a6460f42aa9f998b561582fb4e4cdd6f05dce

1. After deploying the attack contract (0xDDb19a1B), the attacker (0xBDf38B74) proceeded to call the Wooooo function within the attack contract (0xDDb19a1B).

2. The attack contract (0xDDb19a1B) called the function 0x0031b016 of the target contract (0x79cdFd7B) during the attack.

3. The function directly transferred the user's USDT tokens to the attack contract.

2. Vulnerability Analysis

The target contract (0x79cdFd7B) that was attacked is a deprecated HashFlow contract, which was abandoned in May of the previous year and was not open-source. Through reverse engineering, it can be observed that the contract transfers tokens from the “from” address to the “to” address. Based on analysis, it is highly likely that users had granted significant authorization to this contract before May of the previous year. However, after the contract was deprecated, these authorizations were not revoked, and due to potential issues with the restriction logic after deprecation, attackers were able to call functions in the deprecated contract to transfer user assets.

```
if ('cd', 4).length:
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 581] = addr(('cd', 4))
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 613] = addr(('cd', 4))
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 645] = ('cd', 4)
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 545] = 100
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 581 len 28] = addr(('cd', 4)) << 64
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 577 len 4] = transferFrom(address from, address to, uint256 tokens)
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 677] = 32
    mem[ceil32(ceil32(cd[(cd('cd', 4) + 4)])) + 709] = 'SafeERC20: low-level call failed'
```

3. Subsequent Developments

After carrying out the attack, the attacker (0xBDf38B74) open-sourced the attack contract and left a message stating, “Before use recover, please revoke first. Your funds are not safe.” This message serves as a reminder to users to revoke their authorizations to the targeted contract (0x79cdFd7B) before transferring their funds elsewhere.

```
function recover(address token) external payable {
    uint256 amount = userMap[token][msg.sender];
    userMap[token][msg.sender] = 0;
    IERC20(token).safeTransfer(
        msg.sender,
        amount
    );
}

function recoverWithDonate(address token) external payable {
    uint256 amount = userMap[token][msg.sender];
    userMap[token][msg.sender] = 0;
    uint256 transferAmount = amount * 90 / 100;
    uint256 donate = amount - transferAmount;
    IERC20(token).safeTransfer(
        msg.sender,
        transferAmount
    );
    IERC20(token).safeTransfer(
        owner,
        donate
    );
```

The hacker left behind two functions. One function allows users to withdraw all their funds, while the other function leaves 10% of the assets as a reward for the attacker. Currently, users have started withdrawing their funds one by one.

0x116068373 44b1fdd...	Recover	17482215	1 hr 29 mins ago	bayc1333.eth	IN	Hashflow Exploit Contr...	0 ETH	0.0013091
0xf4c6a15b84e170b78...	Recover	17481672	3 hrs 19 mins ago	0x159244...82188BcA	IN	Hashflow Exploit Contr...	0 ETH	0.00053028
0x238f8c4c14427fc34...	Recover	17481414	4 hrs 11 mins ago	riceshower.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00208167
0x755706234ed78298...	Recover	17481392	4 hrs 16 mins ago	riceshower.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00049567
0x8661be41821b668a...	Recover	17481253	4 hrs 45 mins ago	kzsun.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00155067
0xa8a58babc99c02cfff...	Recover	17481244	4 hrs 47 mins ago	kzsun.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00076545
0x7637642ae3a89493...	Recover	17481241	4 hrs 47 mins ago	kzsun.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00152207
0xa96955ac2f840f688...	Recover	17480923	5 hrs 52 mins ago	0x5f5399...4523c04e	IN	Hashflow Exploit Contr...	0 ETH	0.00166215
0xcad5921862cd196af...	Recover	17480178	8 hrs 22 mins ago	awwwkwaarddd.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00247902
0x54c4fcc96f754e560...	Recover	17480055	8 hrs 47 mins ago	moksal.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00179309
0x97d3fcc424dd90a15...	Recover	17480041	8 hrs 50 mins ago	tokenvault.blockchain	IN	Hashflow Exploit Contr...	0 ETH	0.00168718
0x6f08e1185e0ad8170...	Recover	17480024	8 hrs 53 mins ago	gazua.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00122681
0x81ca06fe901fda605...	Recover	17479838	9 hrs 31 mins ago	0xE1a626...3cD7CD21	IN	Hashflow Exploit Contr...	0 ETH	0.00133999
0x471d2c7320a8dc32...	Recover	17479761	9 hrs 46 mins ago	bobchien.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00196735
0xfdd68d61349e67609...	Recover	17479747	9 hrs 49 mins ago	yudaperkasa.eth	IN	Hashflow Exploit Contr...	0.001 ETH	0.00169742
0x4dd39514f56b70ebe...	Recover	17479744	9 hrs 50 mins ago	killacam.eth	IN	Hashflow Exploit Contr...	0 ETH	0.00183211

4. Security Recommendations

The occurrence of this incident was due to the fact that the targeted contract (0x79cdFd7B) had received significant user authorizations in the past, and these authorizations were not revoked after its deprecation, resulting in user asset losses. To prevent similar attacks in the future, it is important to follow these precautions during the development process:

(1) Project developers should thoroughly validate and address any potential logic issues that may arise after deprecating a contract.

(2) Users should regularly review their account authorizations for different protocol contracts and promptly revoke authorizations for contracts they no longer interact with or have been upgraded.

(3) Before deploying contract upgrades, it is crucial to collaborate with professional third-party auditing teams to ensure security.

SharkTeam

In Math, We Trust!

https://sharkteam.org

https://t.me/sharkteamorg

https://twitter.com/sharkteamorg